



TITLE:

制約ベース型初期設計支援システム (数式処理における理論と応用の研究)

AUTHOR(S):

沢田, 浩之

CITATION:

沢田, 浩之. 制約ベース型初期設計支援システム (数式処理における理論と応用の研究). 数理解析研究所講究録 2000, 1138: 211-219

ISSUE DATE:

2000-04

URL:

<http://hdl.handle.net/2433/63814>

RIGHT:

制約ベース型初期設計支援システム

機械技術研究所 沢田 浩之(Hiroyuki Sawada) *

概 要

In engineering design, a lot of design solutions are generated and more and more design parameters are defined when a design progresses. As more design parameters come into design consideration, designers are facing increasing difficulties in gaining an insight to the relationship among these parameters and it results in less optimal design solution. In order to overcome such difficulties, a design support system based on generic constraint solving technique is developed. The system has the following advantages: kinematic and energetic constraints about whole the product are automatically obtained only by connecting basic components selected from the component library, and an incomplete design solution can be appropriately evaluated.

1 はじめに — 初期設計過程 —

与えられた要求仕様をもとに新しく製品を設計する場合、その初期段階における作業は、通常、以下の手順にしたがって行われる。

1. 必要機能の設定

与えられた要求仕様をもとに必要な機能を設定する。要求仕様とは、製品が達成すべき目的を意味する。また、機能とは、その目的を実現するための手段を意味する。例えば、要求仕様が「箱を移動させる」ことであるとき、機能としては「掴んで運ぶ」、「引きずって動かす」などが考えられる。

2. 物理的実体の決定

必要機能を実現する具体的なモノを決定する。例えば、「掴んで運ぶ」という機能を実現する物理的実体として、ロボットハンドなどが考えられる。

3. 条件式の記述

物理的実体が満足すべき条件を、数式として表現する。これらの条件として、幾何拘束や力の釣り合い条件などが挙げられる。

*hiroyuki.sawada@mel.go.jp

4. 設計変数値の計算

条件式を満足するような設計変数値を計算する。

これらの作業過程のうち、「3. 条件式の記述」および「4. 設計変数値の計算」には以下のような問題がある。

(a) 条件式の記述に関する問題

既存製品の手直しではなく新しく製品を設計する場合、その条件式は手書きのラフスケッチなどをもとに手作業で導かれることが多い。例えばロボットアームを設計する場合には、アームを線分で表現した簡単な図を描いて幾何拘束条件を導いたり、また、働く力を表現する矢印などを書き加えて力の釣り合い条件を導くなどといったことが行われる。このような方法では、作業そのものが設計者にとって大きな負担となるばかりでなく、誤解や勘違いに基づく人為的なミスを排除することがきわめて困難である。

(b) 設計変数値の計算に関する問題

設計変数値の計算は、通常、記述された条件式をもとにプログラムを作成し、これを実行することによって行われる。そのプログラムは、解析目的が異なるごとに異なったものが要求される。例えば、入力変数を与えて出力変数の値を求める場合（順問題）と、逆に、出力変数の値を与えて入力変数の値を求める場合（逆問題）とでは異なるプログラムを作成する必要がある。また、通常のプログラムでは入力変数のすべてに具体的な数値を与える必要があるが、設計の初期段階では、値が未定の設計変数が残されていることが多い。従来の方法では、値が未定の入力変数を含んだ式の扱いに困難がある。

さらに、設計解として異なった構造を検討する場合には、上述の作業をその解ごとに行う必要がある。例えば、2 関節アームロボットと 3 関節アームロボットとでは幾何拘束も力の釣り合い式も異なるため、それぞれ別個に条件式を記述し、適正な設計変数値を計算しなくてはならない。これらの作業は設計者にとって大きな負担となるため、様々な異なる解について十分な検討がなされないまま、最初に見つかった解で妥協してしまうことが少なくない。

本研究は、制約評価技術を応用することによってこれらの問題点を解決し、設計者の作業負担を軽減するソフトウェアシステムを構成することを目的とする。本論文の構成は以下の通りである。まず、次章において、本研究における問題解決のアプローチを示す。次に第 3 章でシステムの構成および実装について述べ、第 4 章では例題として 2 関節 2 本指ロボットの設計問題を取り上げる。

2 問題解決の手法

本章では、前章で提示した2つの問題、条件式の記述に関する問題および設計変数値の計算に関する問題について、その解決手法を提案する。

2.1 要素ライブラリ — 条件式記述に関する問題解決手法 —

機械設計では、それが新規設計であっても、すべての構成部品をまったくのゼロから設計することは事実上ありえない。むしろ、既存の部品を組み合わせることによって新しい構造を作り上げることが通常である。条件式の定義／生成という観点から見た場合、これは基本要素に関する局所的な制約条件集合に、要素間の接続条件を追加するものとして捉えられる。ここで提案する要素ライブラリとは、機械設計において多く用いられる基本的な構成部品について、その制約条件式集合を登録したデータベースである。設計者は、要素ライブラリから必要な基本要素を取り出し、それら基本要素間の接続条件を追加することにより、製品全体に関する制約条件集合を得ることができる。要素ライブラリに登録さ

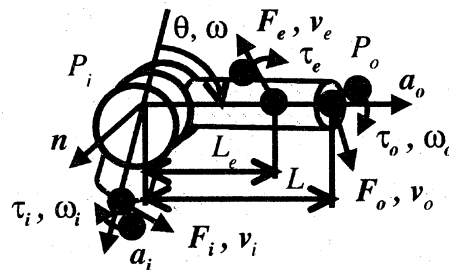


図 1: リンク要素

れている要素の一例として、図1のようなリンク要素を示す。その制約条件は以下のとおりである。

リンク要素に関する制約条件

幾何拘束 $|a_i| = |a_o| = |n| = 1, a_i \times a_o = n \sin \theta, a_i \cdot a_o = -\cos \theta, \overrightarrow{P_i P_o} = L a_o$.

リンク軸回転速度と相対回転速度の関係 $\omega_i \cdot n - \omega_o \cdot n = \omega$

リンク軸回転速度の伝達 $\omega_o \times n = \omega_i \times n$

リンク軸の剛体条件 $v_e - L_e \omega_o \times a_o = v_o - L \omega_o \times a_o = v_i$

力の釣り合い $F_i + F_o = F_e$

トルクの釣り合い $\tau_i + \tau_o - F_o \times L a_o = \sum (\tau_e - F_e \times L_e a_o)$

変数リスト a_i, a_o : リンク軸方向ベクトル、 n : リンク回転軸方向ベクトル、

P_i : リンク回転軸位置、 P_o : リンク先端位置、 L : リンク長、 L_e : 荷重点位置、

θ : リンク軸相対回転角、 ω : リンク軸相対回転速度、 ω_i, ω_o : リンク軸回転速度、
 v_i, v_o : リンク回転軸及びリンク先端における速度、 v_e : 荷重点における速度、
 τ_i, τ_o : 生成されるトルク、 τ_e : 外力トルク、 F_i, F_o : 生成される力、 F_e : 外力。

また、要素ライブラリから抽出された基本要素間のインタフェース条件を設定することによって、製品全体の制約条件集合が得られる例を図 2 に示す。この例では、要素ライブラ

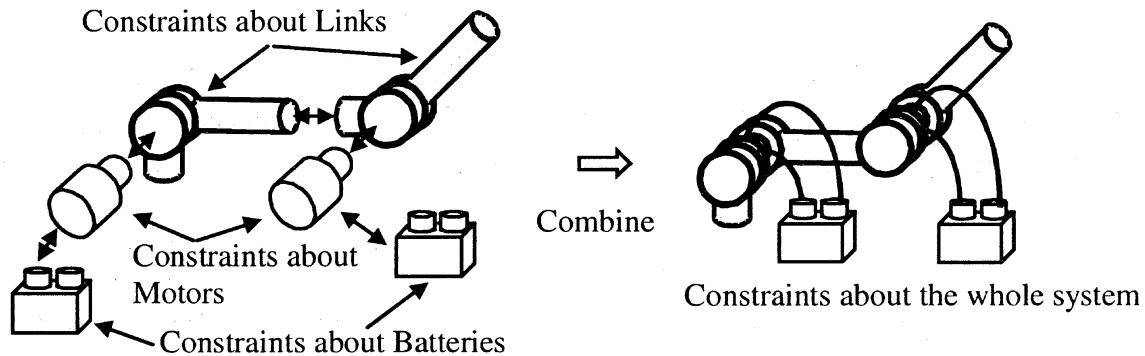


図 2: 制約条件集合の構成

リから、リンク、モータ、電源を選択して組み合わせることによって、ロボットアームが構成されている。設計者は、リンク同士を接続する点の位置座標の一致、リンク軸方向および回転軸方向の一致といった幾何的条件のほか、接続点における力の作用反作用といった力学的条件などをインタフェース条件として定義すればよい。

2.2 代数制約評価系 — 設計変数値計算に関する問題解決手法 —

設計の初期段階では、設計変数の数に比べて制約条件の数が少ないことが多い。これはいわゆる過少制約問題となり、設計変数の値を特定することや、制約条件間の整合性をチェックすることが困難である。また、設計者は、順逆力学問題解析や最適化など様々な解析を必要とするため、システムにはそれらの多様な要求に応えられるだけの柔軟性が求められる。代数制約評価系 [2] は、これらの課題を解決するために開発されたもので、以下の機能を提供する。

1. 不等式間の矛盾検出
2. 矛盾解消のために修正すべき設計変数の特定
3. 具体的な数値解の例示
4. 最適化計算

5. グラフ描画

3 システム構成

図 3 にシステムの構成を、また、図 4 にシステムの概観を示す。システム本体は Visual C++、代数制約評価系は Risa/Asir [1] Windows 版で実装されている。

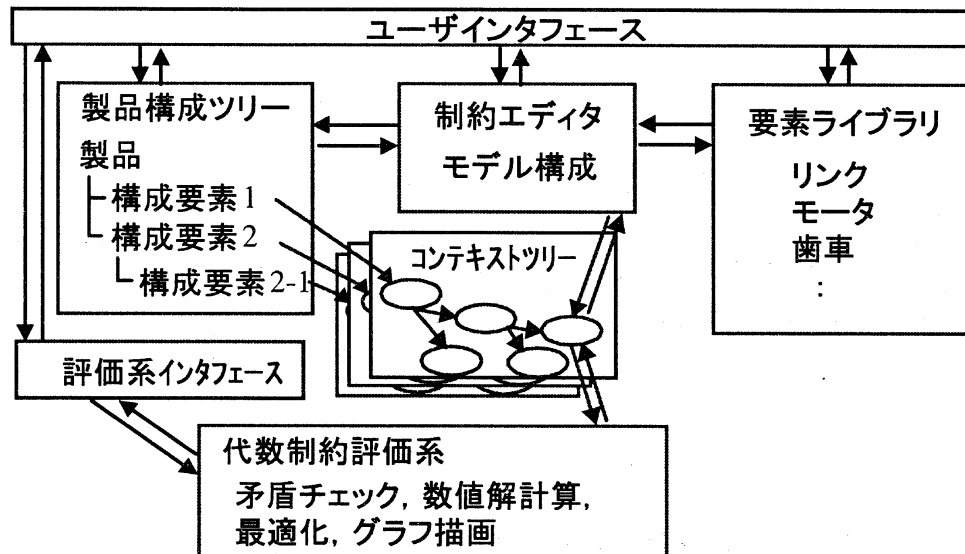


図 3: システム構成

設計者は、ユーザインタフェースを介して操作を行う。まず、製品構成ツリーにおいて、製品全体を構成要素に分割する。大規模な製品を設計する際には、製品全体をいくつかの構成要素に分割したのちそれらを独立に設計し、最後にそれら構成要素を組み合わせることで製品全体を構成するといった方法が多く用いられる。製品構成ツリーは、そのような作業環境を提供するものである (図 4 左上)。

次の作業は製品モデルの構成であり、これは制約エディタ上で行われる。すでに第 2.1 節で述べたように、設計者は必要な部品を要素ライブラリから抽出し、それら部品間の接続条件を定義することによって製品モデルを構成する。今回インプリメントされたシステムでは、要素ライブラリからの部品抽出は、部品アイコンのドラッグ&ドロップによって行われる。また、抽出された部品間に線を引くことによってそれらの間に接続条件が存在することが示される (図 4 右下)。

構成された製品モデルはすべてコンテキストツリーに登録される。設計過程では、構造の変更や寸法の修正などにともない、何種類もの製品モデルが構成される。これらの製品モデルをその制約条件集合の包含関係に基づいて、木構造として分類/整理するものがコ

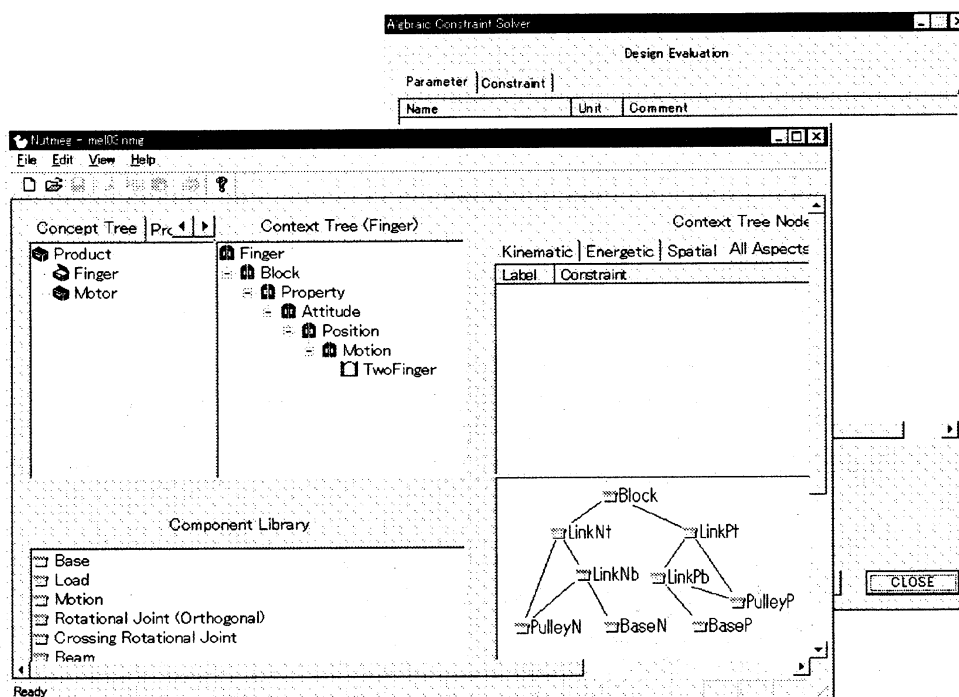


図 4: システム概観

ンテキストツリーである。コンテキストツリーは、製品構成ツリーで分割された各構成要素ごとに1つずつ生成される(図4中央やや左)。ある程度具体化された製品モデルは代数制約評価系で評価される(図4奥)。設計者はその評価結果を参照することにより、さらに製品モデルの改善を進める。

4 例題 — 2関節2本指ロボットの設計 —

例題として、立方体を把持するワイヤ駆動式2関節2本指ロボットの設計を取り上げる(図5)。要求仕様および仮定を以下のように与えるものとする。

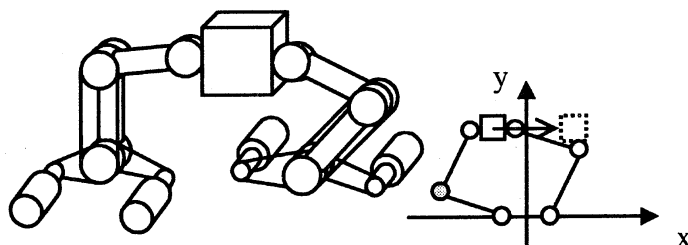


図 5: 2関節2本指ロボット

1. 要求仕様

立方体サイズ $20 [\text{mm}] \times 20 [\text{mm}] \times 20 [\text{mm}]$

立方体質量 $50 [\text{g}]$

立方体重心の動作範囲 $-60 [\text{mm}] \leq x \leq 60 [\text{mm}], y = 80 [\text{mm}]$

立方体の x 方向移動速度 $100 [\text{mm}/\text{sec}]$

2. 仮定

- 立方体とロボット指先間の静止摩擦係数は 0.5 とする。
- ロボットの指を構成するリンクの長さは $60 [\text{mm}]$ 、質量は $200 [\text{g}]$ とする。
- ロボットの指の付け根の位置は、 $x = \pm 20 [\text{mm}]$ とする。
- 図 5 において灰色で示した関節のみを対象とし、そのモータの選択および適正な減速比の決定を行うものとする。

設計作業は以下の手順で行われる。

1. 製品モデル構成
2. 指先の力決定
3. 関節最大出力の計算
4. モータ選定
5. 適正な減速比の決定
6. 解の検証

各作業の詳細について述べる。

1. 製品モデル構成

要素ライブラリから必要な要素を選択し、制約エディタ上で製品モデルを作成する (図 4 右下参照)。製品モデルは、構成要素間の依存関係を示すグラフとして表現される。必要に応じて、寸法や要素間の接続条件を定義する。システムの内部では、自動的に製品全体に関する制約条件集合が構成される。

2. 指先の力の決定

制約評価系に、立方体重心位置とロボット指先の力の関係を示すグラフを描画させ

る。このとき、ロボット指先の力は一意に決まらないものの、そのとりうる値の範囲が不等式として求められる。この場合、次のような出力が得られる。

$$\begin{aligned} F_x &\geq 490 \text{ [mN]}, \\ F_y &= 245 \text{ [mN]}, \\ v_x &= 100 \text{ [mm/sec]}. \end{aligned}$$

ただし、 F_x , F_y , v_x は、それぞれロボット指先の、 x 軸方向の力、 y 軸方向の力、 x 軸方向の速度である。以降、 $F_x = 490 \text{ [mN]}$ として作業を進めることにする。

3. 関節最大出力の計算

制約評価系に、関節の最大出力を計算させる。この場合、 34.5 [mW] が得られる。

4. モータ選定

求められた最大出力に基づいて、市販品のカタログから必要な出力を持つモータを選定する。ここでは、以下のモータを選んだとする。

最大出力	250 [mW]
定格電圧	3 [V]
トルク定数	1.93 [mNm/A]
内部抵抗	8.7 [Ω]

5. 適正な減速比の決定

制約評価系に関節の最大トルクおよび最大角速度を計算させ、それらを制約条件として追加する。さらにモータの定格電圧、トルク定数、内部抵抗を制約条件として追加する。

制約評価系に、すべての条件を満足する減速比を計算させる。この場合、以下の数値解が得られる。

特に条件を追加しなかった場合の減速比:	1/256
関節出力最大時の消費電力を最小とする減速比:	1/931

6. 解の検証

それぞれの減速比について、立方体重心位置とモータ入力電圧のグラフを制約評価系に描画させ、入力電圧が常に定格電圧以下に収まっているかどうかを確認する。描画されたグラフを図 6 に示す。縦軸はモータ入力電圧、横軸は立方体重心位置である。ハッチングされた領域は、解として認められない禁止領域を示す。この図から、減速比が 1/256 のときは問題がないが、1/931 のときはグラフが禁止領域あるいはその近傍を通ることがわかる。通常、このような場合には減速比として 1/931 を採用することはない。

